```
pragma solidity ^0.5.3;

contract voting{

    address public contractOwner;

    address[] public candidatesList;

    mapping (address => uint8) public votesRecieved;

    address public winner;
    uint public winnerVotes;

    enum votingStatus { NotStarted, Running, Completed }
    votingStatus public status;


    constructor() public{
        contractOwner = msg.sender;
    }

    modifier OnlyOwner{
        if(msg.sender == contractOwner){
            _;
        }
    }
    function setStatus() OnlyOwner public{
       if (status != votingStatus.Completed && status !=
votingStatus.Running ){
            status = votingStatus.Running;
        }else{
            status = votingStatus.Completed;
        }

    }

    function registerCandidates(address _candidate) OnlyOwner public {
        candidatesList.push(_candidate);
    }

    function vote(address _candidate) public{
        require(validateCandidate(_candidate), "Not a Valid
Candidate");

        require(status == votingStatus.Running, "Election is not
active");

        votesRecieved[_candidate] = votesRecieved[_candidate] + 1;
//        votesRecieved[_candidate] += 1;
    }
```

```
    function validateCandidate(address _candidate) view public
returns(bool){

        for(uint i = 0; i < candidatesList.length; i++){
            if (candidatesList[i] == _candidate){
                return true;
            }
        }
        return false;
    }

    function votesCount(address _candidate) public view returns(uint){

        require(validateCandidate(_candidate), "Not a Valid
Candidate");

        assert(status == votingStatus.Running);

        return votesRecieved[_candidate];
    }

    function result() public{
        require(status == votingStatus.Completed, "Voting is not
completed, Result can't be declared");

        for (uint i = 0; i < candidatesList.length; i++){
            if (votesRecieved[candidatesList[i]] > winnerVotes){
                winnerVotes = votesRecieved[candidatesList[i]] ;
                winner = candidatesList[i];
            }
        }
    }
}
```